UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/810,716	03/16/2001	Hiang-Swee Chiang	**GP-0002	2184

23377 7590 04/30/2010

WOODCOCK WASHBURN LLP
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
BULLOCK JR, LEWIS ALEXANDER	
ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
04/30/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte HIANG-SWEE CHIANG

Appeal 2009-001205
Application 09/810,716
Technology Center 2100

Decided: April 30, 2010

Before JOSEPH L. DIXON, ST. JOHN COURTENAY III, and DEBRA K. STEPHENS, *Administrative Patent Judges.*

DIXON, *Administrative Patent Judge.*

DECISION ON APPEAL

The Appellant appeals under 35 U.S.C. § 134(a) from the final rejection of claims 2, 4-29, 31-50, 52-64, 66-78, and 162-169. Claims 1, 3, 30, 51, 65, 79-161, and 170-174 have been canceled. We have jurisdiction under 35 U.S.C. § 6(b).

We Affirm.

I. STATEMENT OF THE CASE

The Invention

The invention at issue on appeal relates to a method, system and apparatus for generating a complete web application source code by reading a set of graphical web application screens (Spec. 1).

The Illustrative Claim

Claim 2, an illustrative claim, reads as follows:

> 2. A method of generating computer code for a web application, comprising:
>
> receiving input files, wherein the input files are at least one web application graphical user interface;
>
> generating an application framework code and an event handler skeleton, wherein generating an event handler skeleton comprises:
>
> parsing at least one input file;
>
> reviewing the parsed input file for one or more of a tag type, an attribute name and an attribute value; and
>
> determining an event handler method based on one or more of the tag type, the attribute name and the attribute value;
>
> receiving web application business logic objects;
>
> receiving event handler methods;

organizing the application framework code, the web application business logic objects and the event handler methods into application source code; and

binding the web application source code with the input files at runtime.

The References

The Examiner relies on the following references as evidence:

Lau	US 5,987,247	Nov. 16, 1999
Quaeler-Bock	US 6,023,271	Feb. 8, 2000
Lindhorst	US 6,337,696 B1	Jan. 8, 2002 (filed May 10, 1999)

The Rejection

The following rejections are before us for review:

Claims 2, 4-29, 31-50, 52-64, 66-78, and 162-169 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Lau, Lindhorst, and Quaeler-Bock.

Only those arguments actually made by the Appellant have been considered in this decision. Arguments which the Appellant could have made but chose not to make in the Briefs have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37 (c)(1)(vii) (2008).

II. ISSUE

Has the Examiner erred in finding that the combination of Lau, Lindhorst, and Quaeler-Bock teaches or fairly suggests "generating an event

handler skeleton comprises . . . determining an event handler method based on one or more of the tag type, the attribute name and the attribute value," as recited in claim 1?

III. PRINCIPLES OF LAW

Scope of Claim

During prosecution before the USPTO, claims are to be given their broadest reasonable interpretation, and the scope of a claim cannot be narrowed by reading disclosed limitations into the claim. *See In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997). The Office must apply the broadest reasonable meaning to the claim language, taking into account any definitions presented in the specification. *In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004) (citing *In re Bass*, 314 F.3d 575, 577 (Fed. Cir. 2002)).

"Giving claims their broadest reasonable construction 'serves the public interest by reducing the possibility that claims, finally allowed, will be given broader scope than is justified.'" *Id.* (quoting *In re Yamamoto*, 740 F.2d 1569, 1571 (Fed. Cir. 1984)). "Construing claims broadly during prosecution is not unfair to the applicant . . . because the applicant has the opportunity to amend the claims to obtain more precise claim coverage." *Id.*

Obviousness

"Obviousness is a question of law based on underlying findings of fact." *In re Kubin*, 561 F.3d 1351, 1355 (Fed. Cir. 2009). The underlying factual inquiries are: (1) the scope and content of the prior art, (2) the differences between the prior art and the claims at issue, (3) the level of ordinary skill in the pertinent art, and (4) secondary considerations of nonobviousness. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007) (citation omitted).

IV. FINDINGS OF FACT

The following findings of fact (FFs) are supported by a preponderance of the evidence.

Specification

1. Appellant's Specification describes an event handler as follows:

> Event handler code 620 comprises a *skeleton or framework* made up of an event handler (click method) object that accepts requests from HTML <input> tags with JLMClick name and submit type attributes in the HTML files . . . web developers 215 simply need to write appropriate source code into the event handler code skeleton to drive the application business logic in response to user interactions.

(Spec. 18, ll. 3-11, emphasis added).

Lindhorst

2. Lindhorst discloses that an event handler script is recognized (generated) by scanning or parsing an indexed HTML text such as:

1.<SCRIPT>

. . . randomcode . . .

sub button2_ dbclick . . . eventhandlercode . . . end sub

. . . randomcode . . .

</SCRIPT>

(col. 15, ll. 60-64).

3. Lindhorst further discloses the process of generating a script for an event handler:

> If a script start tag was not found at the decision state 334 then the process 110 moves to a decision state 340 wherein a determination is made whether the tag discovered at the decision state 314 is an HTML start tag. If an HTML start tag is found at the decision state 340, the process 110 moves to a decision state 344 wherein *a determination is made whether the HTML start tag is a scriptable tag. As discussed above, a scriptable HTML tag is the type of tag that provides events and actions that can be associated with VBScript JavaScript programs. See Table 1 above for a list of scriptable HTML tags.*
>
> *If a scriptable HTML tag is found at the decision state 344, then the process 110 moves to a state 348 wherein the location of the scriptable HTML tag within the document is saved. The process 110 then advances to a state 350 and moves to the HTML end tag and captures the text between the HTML start and end tags.*

> It should be noted that advances in HTML programming may yield additional types of scriptable tags which may be used within the present invention. Thus, this invention is not limited to providing event handlers for object, script and HTML tags, but can be more generally used to provide scripts between events and actions of many types of objects.

(col. 16, ll. 34-56, Fig. 5, emphasis added).

4. More specifically, Lindhorst further describes an example of generating an event handler method of invocation:

> As an example of a method invocation, consider the invocation of the `Show` method of the `Button` object. *The invocation is again a merging of the names taken from the meta information describing the 'Show' method and the syntactic requirements dictated by the language support code.* The resulting invocation string for VBScript is
>
> call Button.Show()
>
> while the JavaScript support code would generate
>
> Button.Show();
>
> The introduction of "call", the period and the parentheses are the implemented by the language support code.
>
> If there had been parameters on the method, then the names of the parameter would have been taken from the description of the 'Show' method, and the syntax needed to incorporate those names into the method invocation would have been supplied by the language support code. As a brief example, *if the `Show` method included two parameters 'x' and 'y', the resulting invocation strings would have been:*

call Button.Show(x, y) VBScript or

Button.Show(x, y); JavaScript

(col. 20, ll. 9-28, Fig. 6, emphasis added).

5. Lindhorst also discloses that a user can select an object in the event pane, and select an event linked to the object, and then select an action associated to the object in the action pane. After the selections, a script to link an event to an action is generated by the Script Wizard (col. 11, l. 66-col. 12, l. 17).

V. ANALYSIS

The Appellant has the opportunity on appeal to the Board of Patent Appeals and Interferences (BPAI) to demonstrate error in the Examiner's position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) (citing *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)).

The Examiner sets forth a detailed explanation of a reasoned conclusion of unpatentability in the Examiner's Answer. Thus, we look to the Appellant's Brief to show error in the proffered reasoned conclusion. *Id.*

Grouping of Claims

The Appellant has elected to argue claims 2, 4-29, 31-50, 52-64, 66-78, and 162-169 together as a group (App. Br. 10). Therefore, we select independent claim 2 as the representative claim for this group, and we will address the Appellant's arguments with respect thereto. 37 C.F.R. § 41.37 (c)(1)(vii). *See In re Nielson*, 816 F.2d 1567, 1572 (Fed. Cir. 1987).

35 U.S.C. § 103(a) rejections

We begin our analysis with claim construction. Since the claims do not expressly define the term "event handler skeleton," we thus consult the Summary of Claimed Subject Matter (App. Br. 3) and the corresponding portions of the Specification for *context* to determine the meaning and the scope of the disputed claim term. We find that the Specification describes the event handler skeleton as a framework code made up of an event handler object. (FF 1). Therefore, we broadly, yet reasonably construe the claim limitation "event handler skeleton" as a framework or a supporting infrastructure for an event handler script or code. We also broadly, yet reasonably construe the claim limitation "determining . . . based on one or more of the tag type, the attribute name and the attribute value" as any one of tag type, the attribute name and the attribute value which has any nexus or association with the action of recognition, formation, or generation.

With respect to claim 1, the Appellant contends that "Lindhorst does not even mention the concept of an *event handler skeleton*, and certainly does not teach '*generating* an event handler skeleton.'" (App. Br. 11).

We disagree with the Appellant's contention. We find Lindhorst discloses that at least one framework of the event handler scripts is generated (FF 2) under our claim construction, i.e., the script has supporting infrastructure or frame work that has the name of the function, function code (sub routine), and other codes.

The Appellant further contends that Lindhorst only teaches a user selects an event icon, but there is no teaching or suggestion that the user makes the determination based on one or more of the tag type, the attribute name and attribute value (App. Br. 11, Reply Br. 7), and neither cited references teaches the claimed limitation "determining an event handler method based on one or more of the tag type, the attribute name and the attribute value," as recited in claim 1. (App. Br. 14, Reply Br. 4).

We disagree with the Appellant's contentions. We find Lindhorst teaches that in the process of generating an event handling method, the type of the tag is first determined (i.e., *a scriptable HTML tag is the type of tag that provides events and actions that can be associated with VBScript or JavaScript programs*), and then the process captures the text between the starting tag and ending tag to generate the script of the event handling method (FF 3). Furthermore, we find Lindhorst teaches that a two-parameter event handler method, *call Button.Show(x, y)* or *Button.Show(x, y)*, is determined by finding the method has two parameters in the description of the method (FF 4). Finally, we are unpersuaded by Appellant's argument that a user determines the event handler method. We find that the user only selects high level icon to determine whether the next process is to generate a method, etc. (FF 5). The process of generating a method/script such as how many parameters in the method are determined based on the parameters found in the description of the method (FF 4), is read on the claimed language.

Accordingly, we sustain the Examiner's obviousness rejection of claim 1. Independent claims 27, 48, 64, 162, and 163-165 contain similar limitations, and the Appellant presents similar arguments thereto. Therefore, we also sustain the Examiner's obviousness rejection of claims 27, 48, 64, 162, and 163-165. We also sustain the Examiner's obviousness rejections of dependent claims 4-26, 28-29, 31-47, 49-50, 52-63, 66-78, and 167-169 which have not been separately argued, and they fall with their base claims. 37 C.F.R. § 41.37(c)(1)(vii). *See In re Nielson*, 816 F.2d at 1572.

VI. CONCLUSION

Based on our consideration of the totality of the record before us, we have weighed the evidence of obviousness found in the combined teachings of the applied references, with Appellant's countervailing evidence and arguments for nonobviousness and conclude that the claimed invention encompassed by appealed claims 2, 4-29, 31-50, 52-64, 66-78, and 162-169 would have been obvious as a matter of law under 35 U.S.C. § 103(a).

VII. ORDER

We affirm the obviousness rejection of claims 2, 4-29, 31-50, 52-64, 66-78, and 162-169 under 35 U.S.C. § 103(a).

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136 (a). *See* 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

rwk

WOODCOCK WASHBURN LLP
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA PA 19104-2891